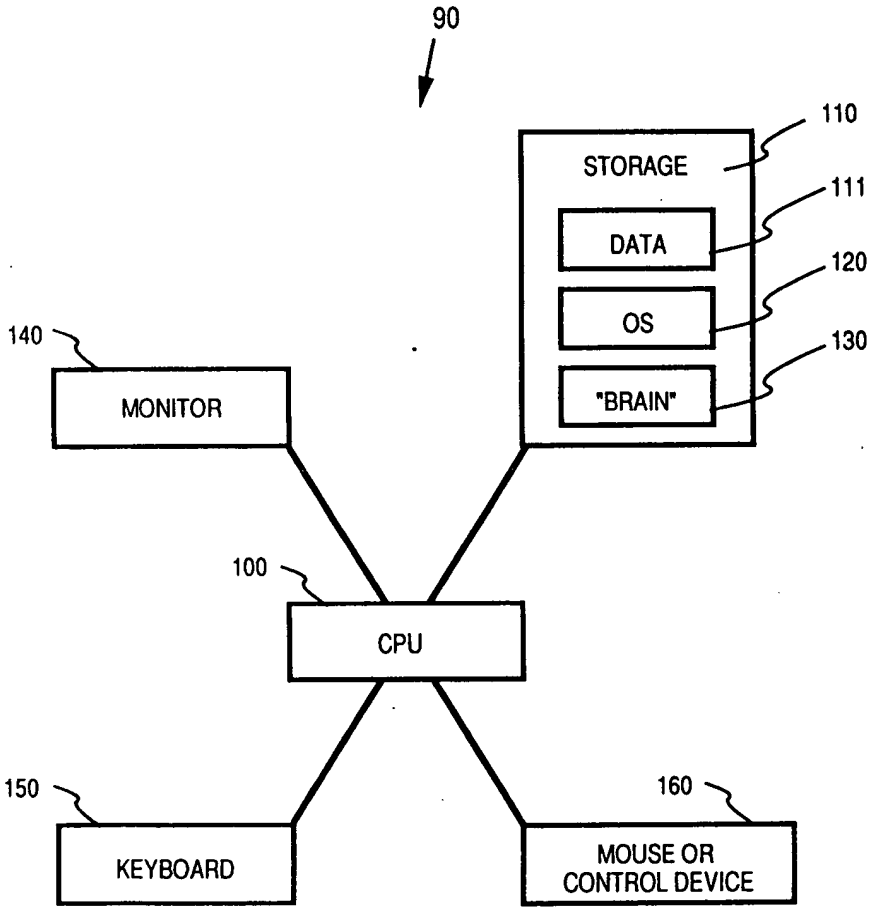

Fig.1

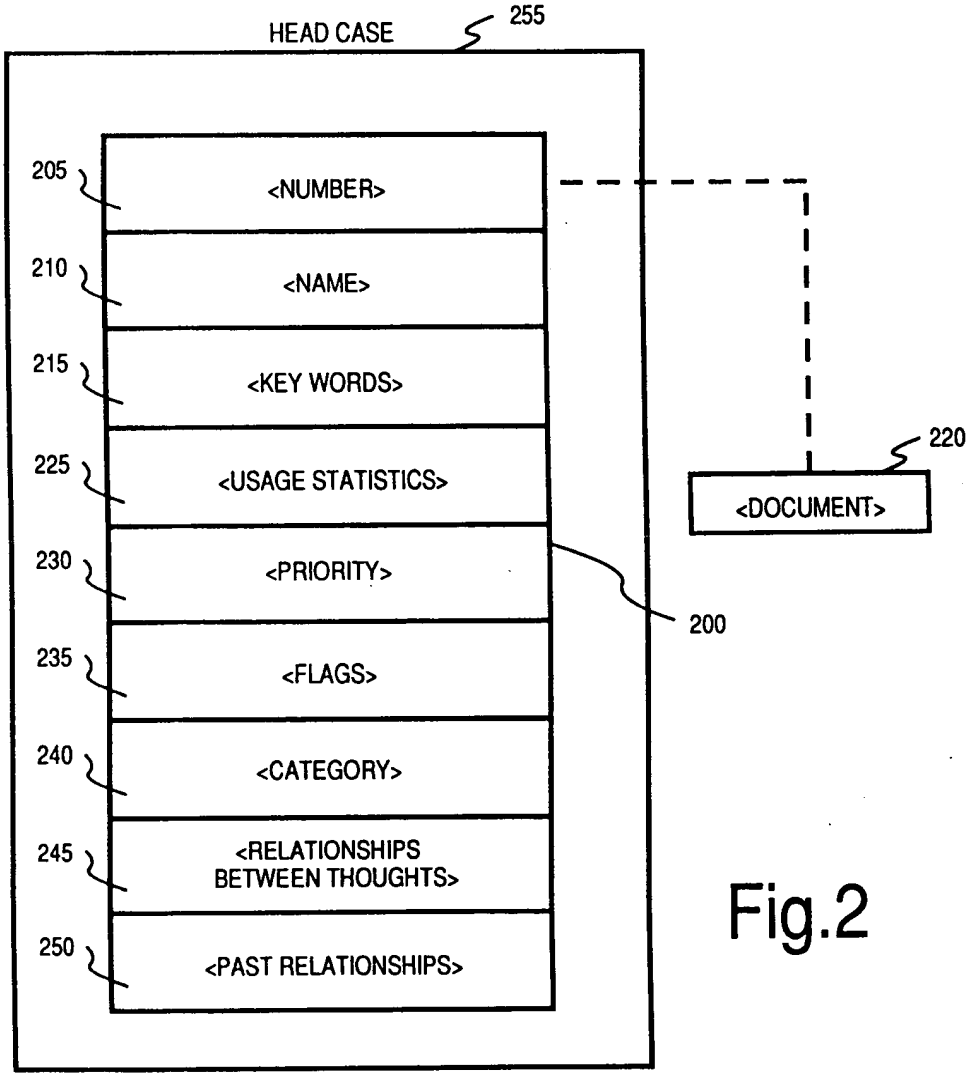
HEAD CASE
255
205
<NUMBER>
210
<NAME>
215
<KEY WORDS>
225
<USAGE STATISTICS>
230
<PRIORITY>
235
<FLAGS>
240
<CATEGORY>
245
<RELATIONSHIPS BETWEEN THOUGHTS>
250
<PAST RELATIONSHIPS>
220
<DOCUMENT>
200

Fig.2

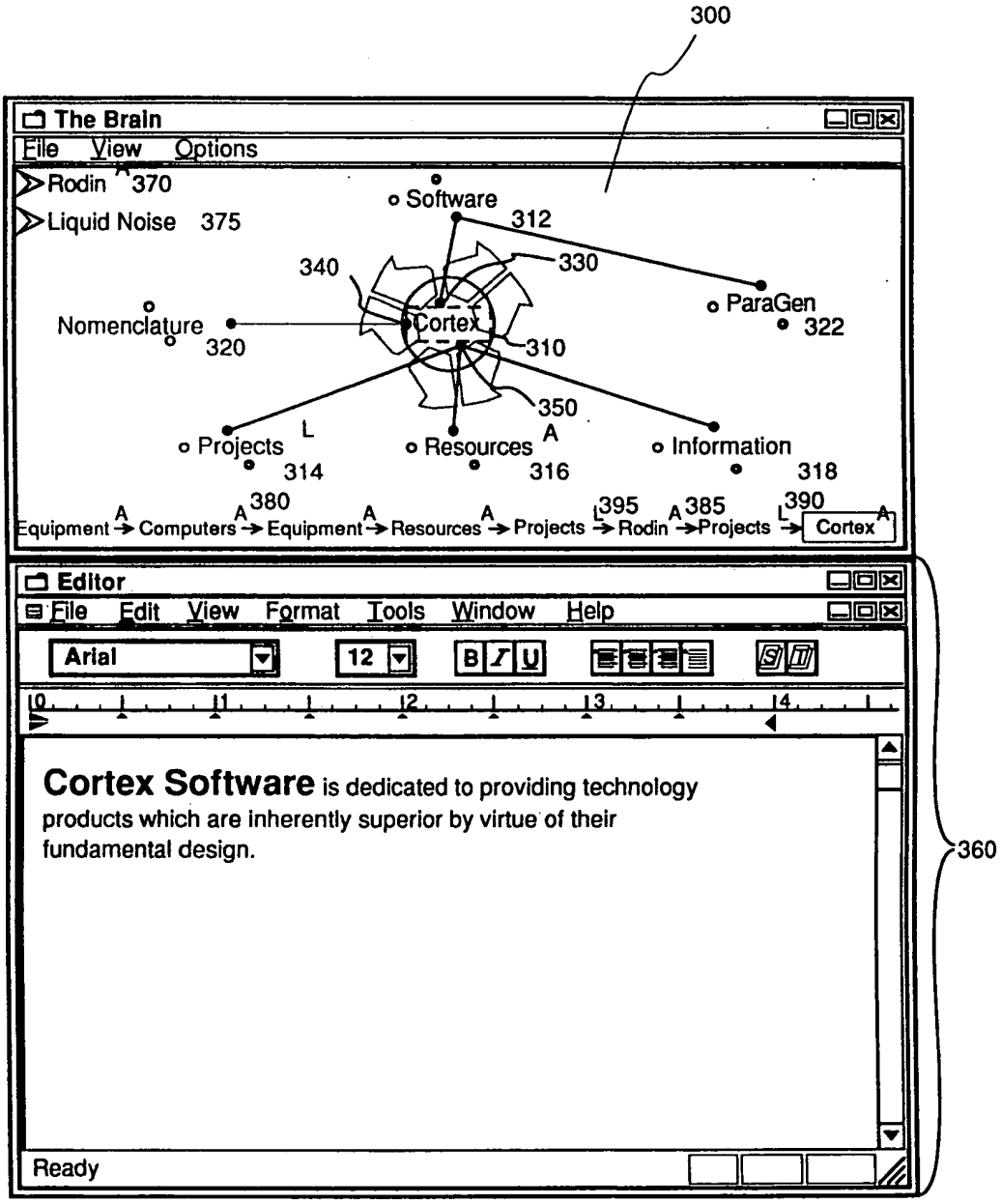
300
The Brain
File View Options
Rodin 370
Liquid Noise 375
Software
312
340
330
Nomenclature
320
Cortex
310
ParaGen
322
350
Projects
314
Resources
316
Information
318
Equipment → Computers → Equipment → Resources → Projects → Rodin → Projects → Cortex
380
395
385
390
Editor
File Edit View Format Tools Window Help
Arial
12
B I U
Cortex Software is dedicated to providing technology products which are inherently superior by virtue of their fundamental design.
Ready
360

Fig. 3

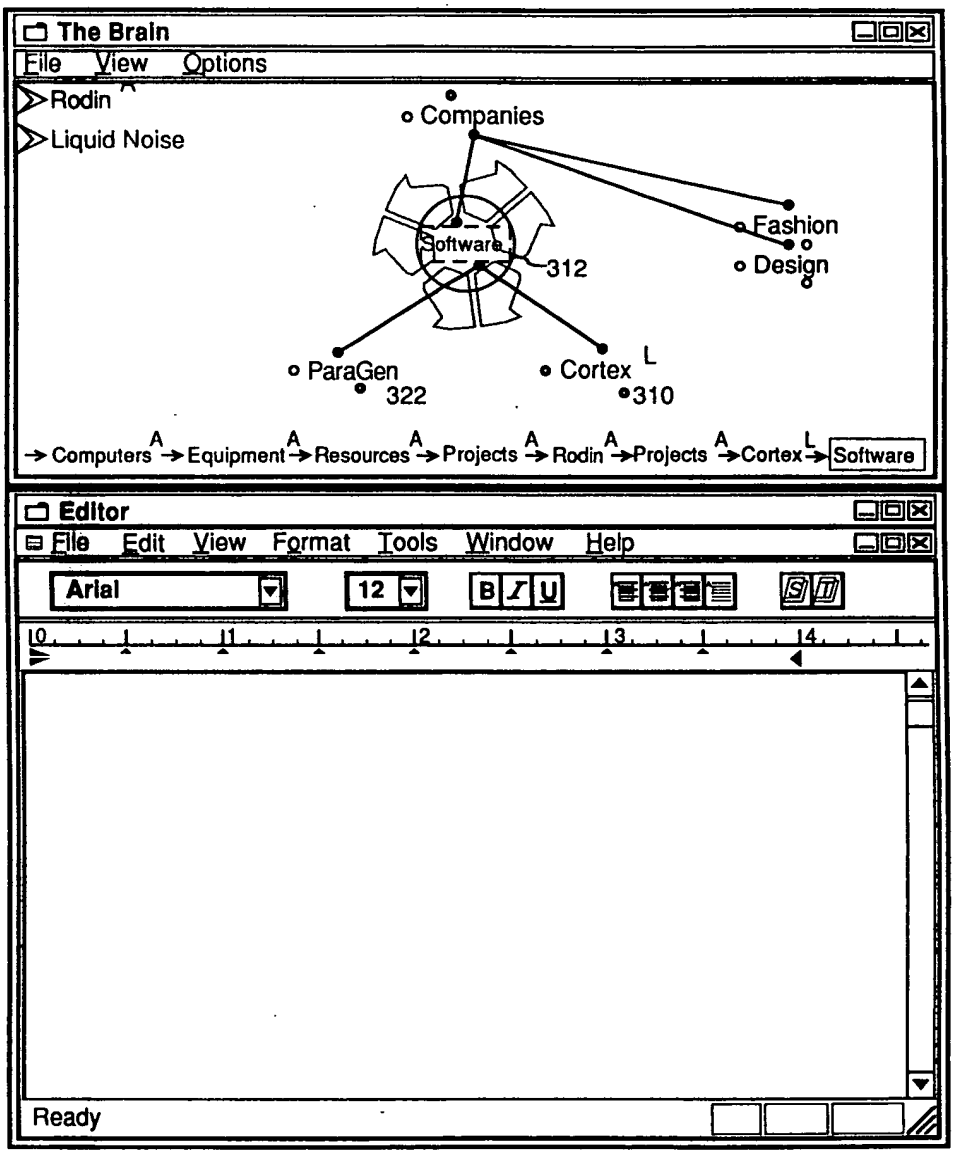
The Brain
File View Options
Rodin
Liquid Noise
Companies
Software
312
Fashion
Design
ParaGen
322
Cortex
310
Computers A Equipment A Resources A Projects A Rodin A Projects A Cortex L Software
Editor
File Edit View Format Tools Window Help
Arial
12
B I U
Ready

Fig. 4

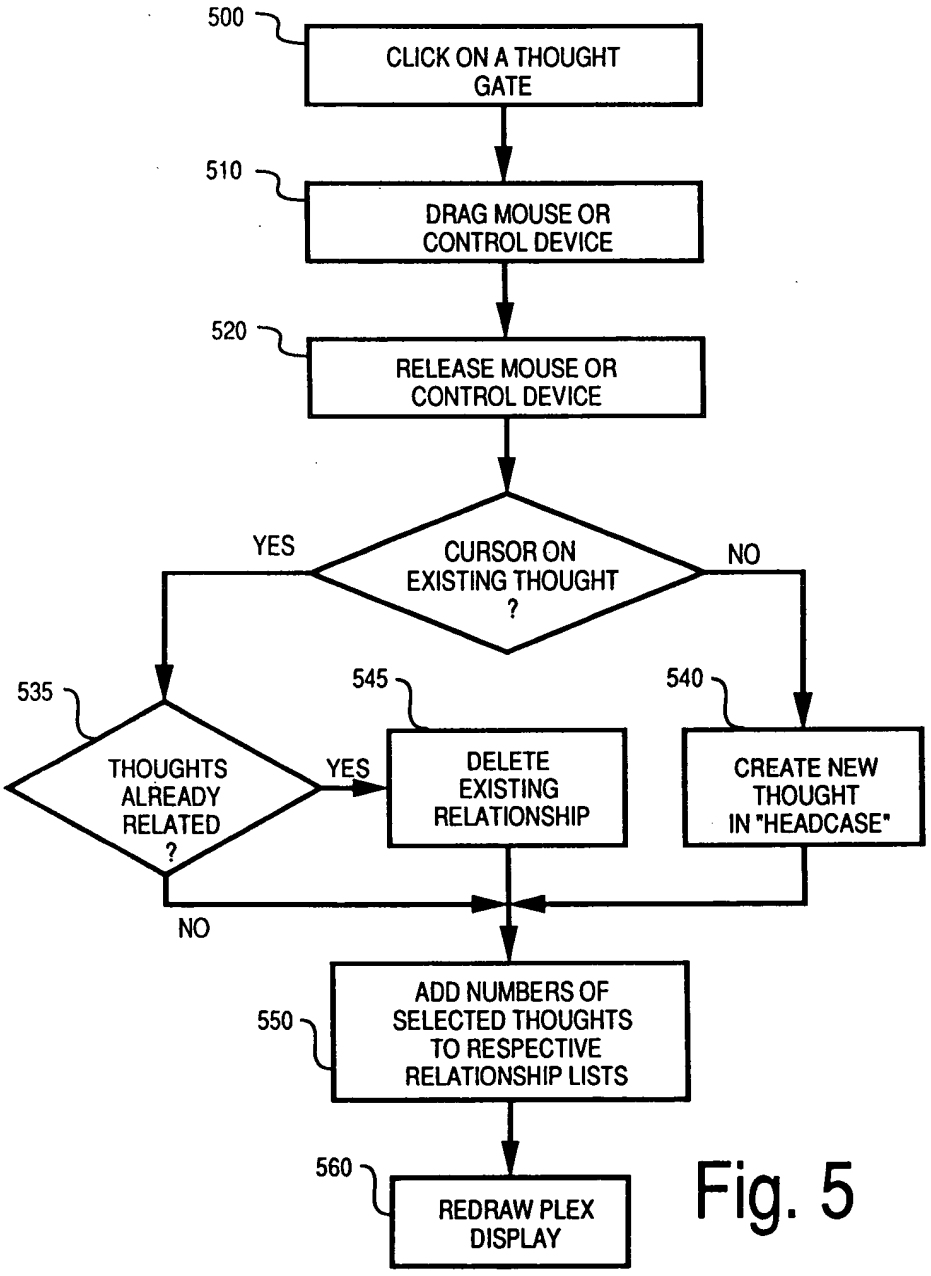
500
CLICK ON A THOUGHT GATE
510
DRAG MOUSE OR CONTROL DEVICE
520
RELEASE MOUSE OR CONTROL DEVICE
CURSOR ON EXISTING THOUGHT ?
YES
NO
535
THOUGHTS ALREADY RELATED ?
YES
NO
545
DELETE EXISTING RELATIONSHIP
540
CREATE NEW THOUGHT IN "HEADCASE"
550
ADD NUMBERS OF SELECTED THOUGHTS TO RESPECTIVE RELATIONSHIP LISTS
560
REDRAW PLEX DISPLAY

Fig. 5

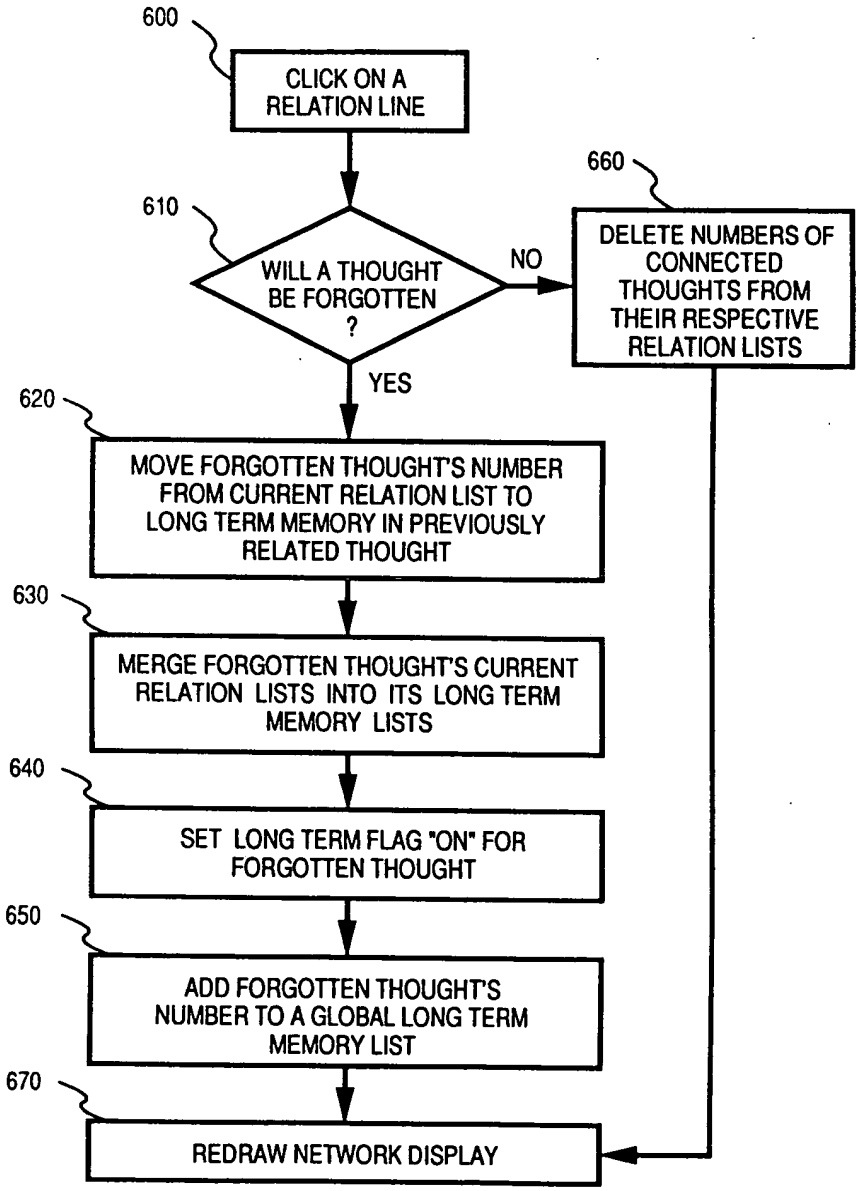
600
CLICK ON A RELATION LINE
610
WILL A THOUGHT BE FORGOTTEN ?
NO
660
DELETE NUMBERS OF CONNECTED THOUGHTS FROM THEIR RESPECTIVE RELATION LISTS
YES
620
MOVE FORGOTTEN THOUGHT'S NUMBER FROM CURRENT RELATION LIST TO LONG TERM MEMORY IN PREVIOUSLY RELATED THOUGHT
630
MERGE FORGOTTEN THOUGHT'S CURRENT RELATION LISTS INTO ITS LONG TERM MEMORY LISTS
640
SET LONG TERM FLAG "ON" FOR FORGOTTEN THOUGHT
650
ADD FORGOTTEN THOUGHT'S NUMBER TO A GLOBAL LONG TERM MEMORY LIST
670
REDRAW NETWORK DISPLAY

Fig. 6

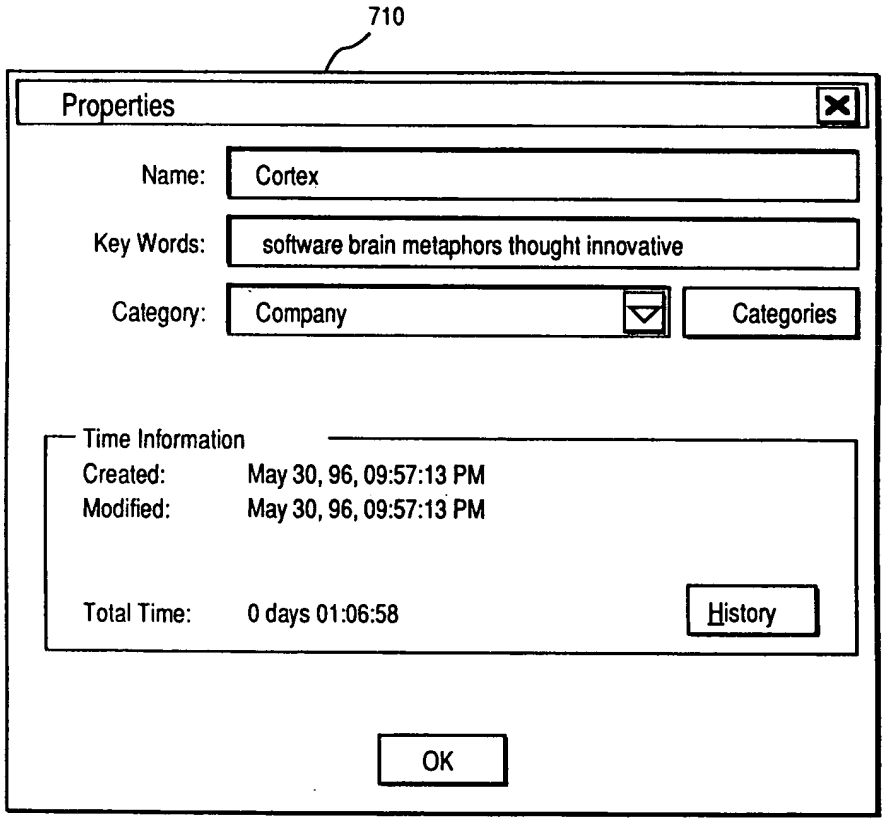
710
Properties
Name:
Cortex
Key Words:
software brain metaphors thought innovative
Category:
Company
Categories
Time Information
Created:
May 30, 96, 09:57:13 PM
Modified:
May 30, 96, 09:57:13 PM
Total Time:
0 days 01:06:58
History
OK

Fig. 7

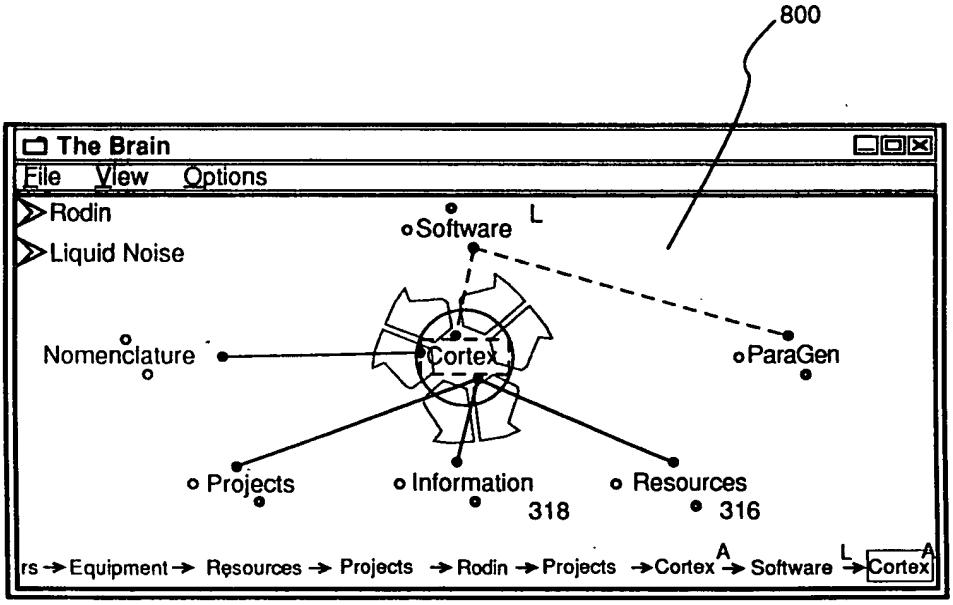
800
The Brain
File View Options
Rodin
Liquid Noise
Software
L
Nomenclature
Cortex
ParaGen
Projects
Information
318
Resources
316
rs → Equipment → Resources → Projects → Rodin → Projects → Cortex → Software → Cortex
A
L
A

Fig. 8

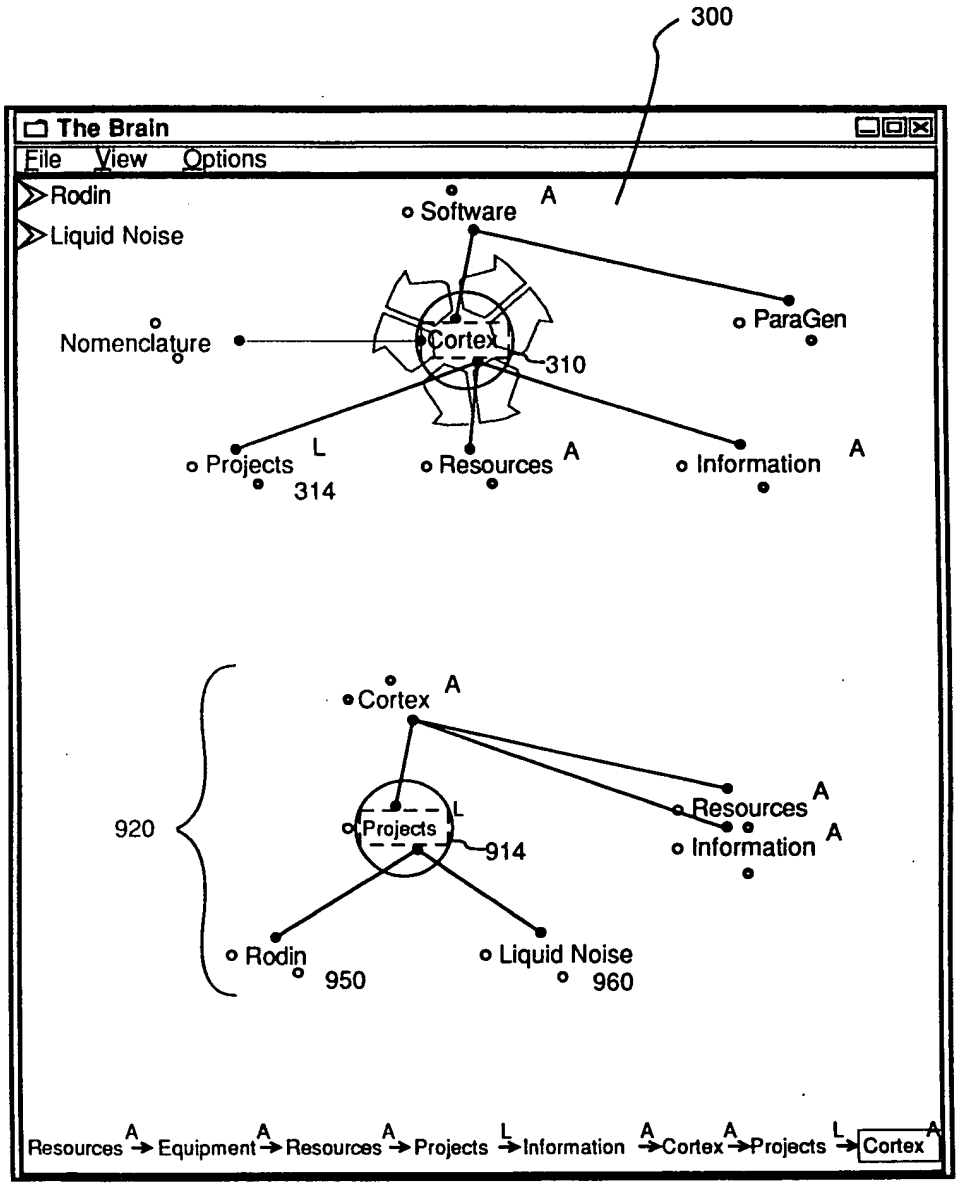
300
The Brain
File View Options
Rodin
Liquid Noise
Software
A
Nomenclature
Cortex
310
ParaGen
Projects
L
314
Resources
A
Information
A
920
Cortex
A
Projects
L
914
Resources
A
Information
A
Rodin
950
Liquid Noise
960
Resources A → Equipment A → Resources A → Projects L → Information A → Cortex A → Projects L → Cortex A

Fig. 9

```
boolean CheckForIsolation(int centralThought, int targetThought)
{
	// this function checks if centralThought is related to targetThought
	// via any of targetThought's relations (not directly)

	// remove centralThought as a direct relation from targetThought
	RemoveRelation(targetThought, centralThought);

	// create an empty thought list to keep track of the search
	intList searchList = CreateEmptyList();

	// start recursive searches on each of targetThought's direct relations
	int relation = GetFirstRelation(targetThought);
	boolean found;
	do {
		found = Search(relation, centralThought, searchList);
		if(found) {
			// centralThought was found, no need to search any further
			break;
		}
		// this loop will end when there are no more relations
	} while(relation = GetNextRelation(targetThought);

	// add centralThought back onto targetThought as a relation
	AddRelation(targetThought, centralThought);

	return found;
}
```

Fig. 10

```
boolean Search(source, dest, searchList)
{
        if(Find(source, searchList)) {
                // source has already been searched
                return FALSE;

        }

        // add source to the searchList
        Add(source, searchList)

        if(source == dest) {
                // this is the destination, we have found it
                return TRUE;

        }

        // recursive searches on each of sources direct relations
        int relation = GetFirstRelation(source);
        boolean found;
        do {
                found = Search(relation, dest, searchList);
                if(found) {
                        // centralThought was found, no need to search any further
                        break;
                }
                // this loop will end when there are no more relations
        } while(relation = GetNextRelation(targetThought);

        return found;
}
```

Fig. 10(cont'd)

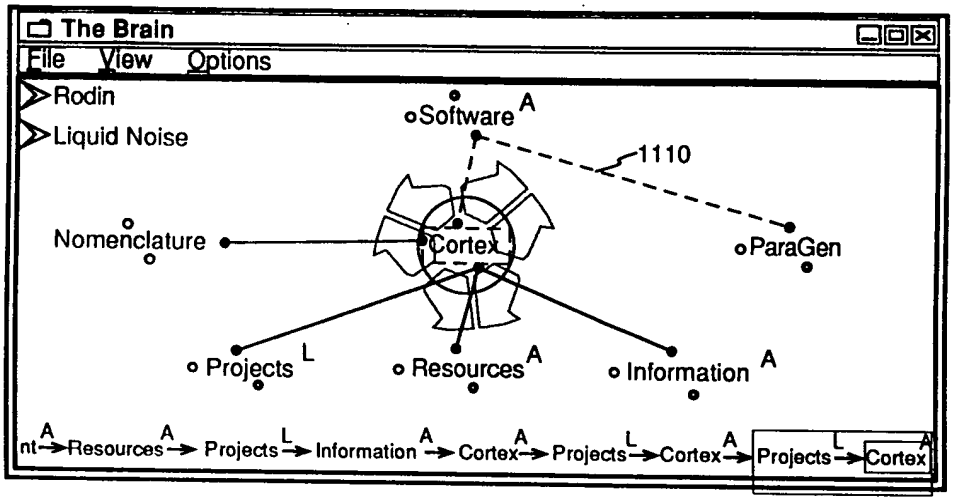
The Brain
File View Options
Rodin
Liquid Noise
Software A
1110
Nomenclature
Cortex
ParaGen
Projects L
Resources A
Information A
nt A → Resources A → Projects L → Information A → Cortex A → Projects L → Cortex A → Projects L → Cortex A
Create Train of Thought
1120

Fig. 11

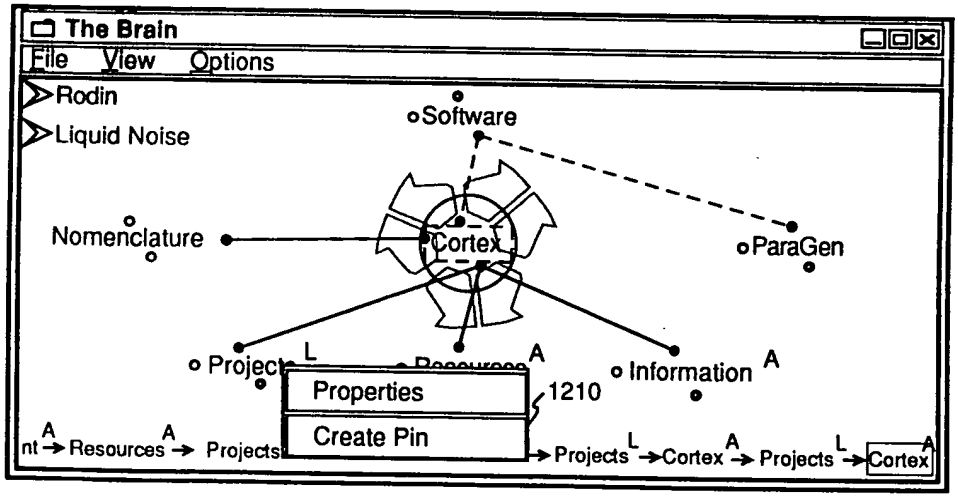
The Brain
File View Options
Rodin
Liquid Noise
Software
Nomenclature
Cortex
ParaGen
Information A
Properties
1210
Create Pin
nt A → Resources A → Projects
Projects L → Cortex A → Projects L → Cortex A

Fig. 12

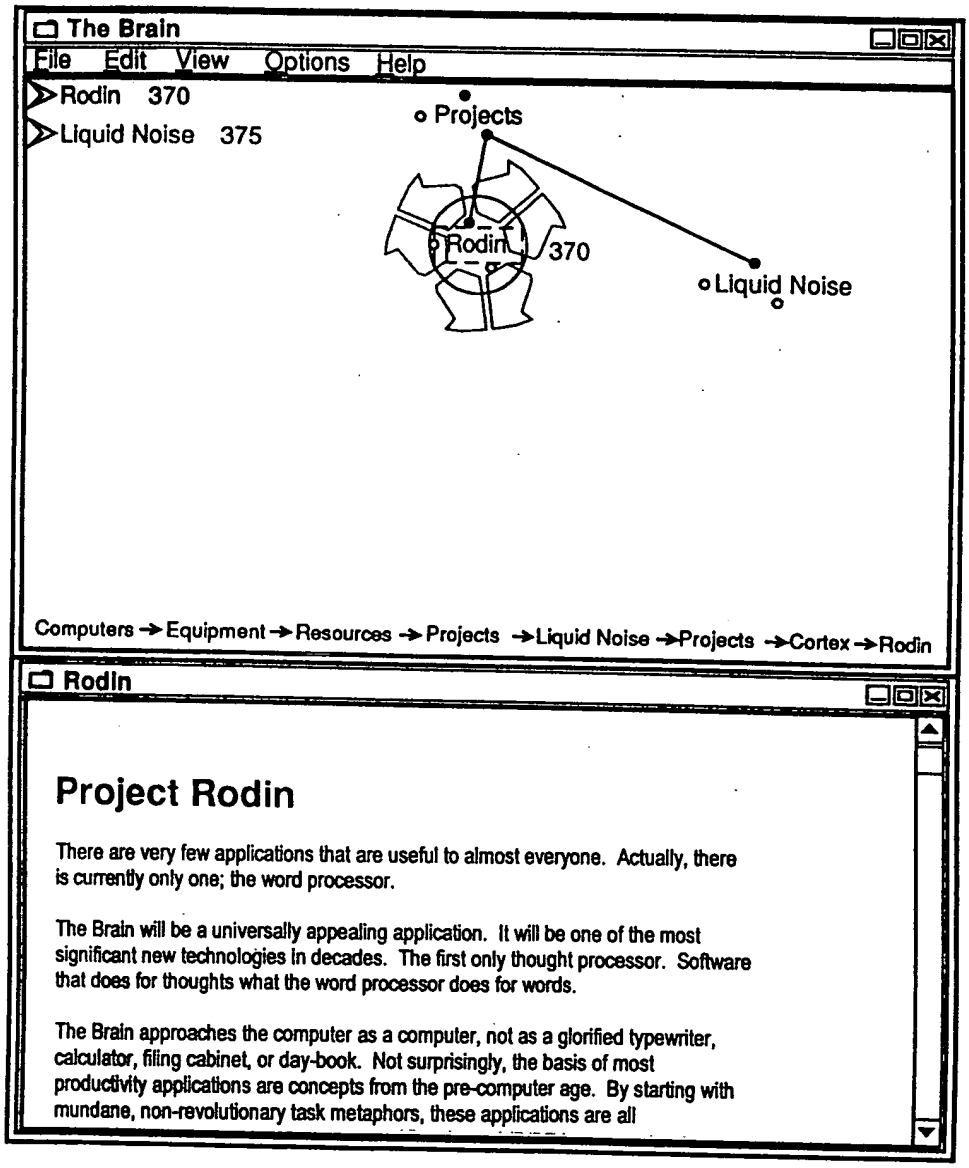
The Brain
File Edit View Options Help
Rodin 370
Liquid Noise 375
Projects
Rodin
370
Liquid Noise
Computers → Equipment → Resources → Projects → Liquid Noise → Projects → Cortex → Rodin
Rodin
Project Rodin
There are very few applications that are useful to almost everyone. Actually, there is currently only one; the word processor.
The Brain will be a universally appealing application. It will be one of the most significant new technologies in decades. The first only thought processor. Software that does for thoughts what the word processor does for words.
The Brain approaches the computer as a computer, not as a glorified typewriter, calculator, filing cabinet, or day-book. Not surprisingly, the basis of most productivity applications are concepts from the pre-computer age. By starting with mundane, non-revolutionary task metaphors, these applications are all

Fig. 13

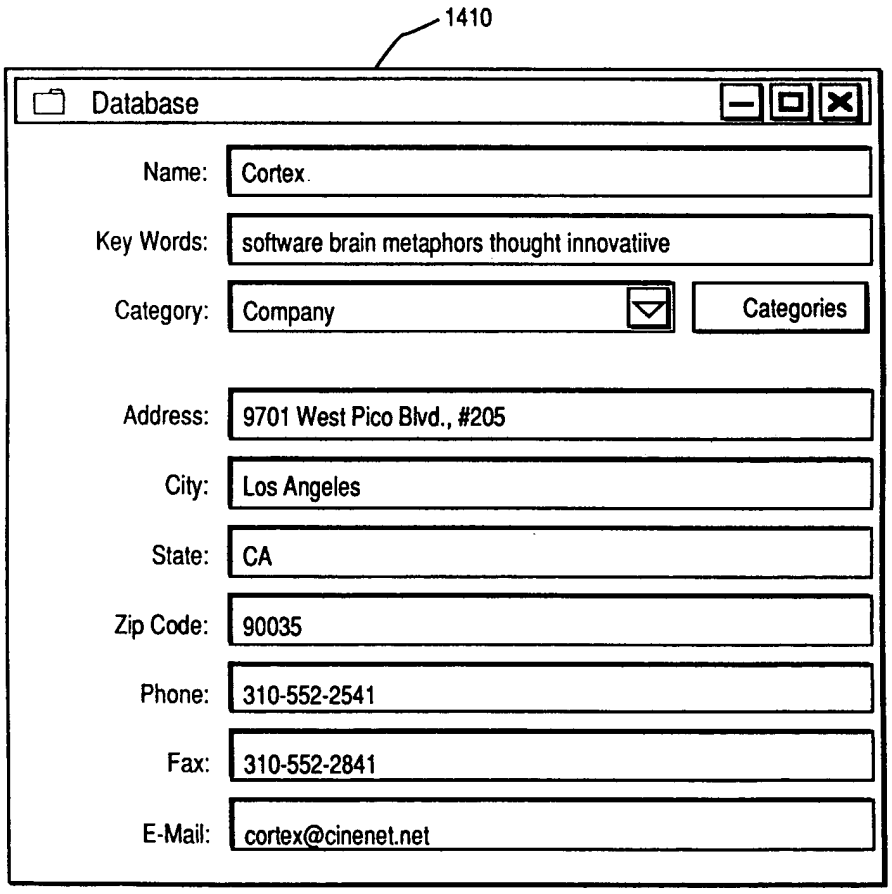
1410
Database
Name:
Cortex
Key Words:
software brain metaphors thought innovatiive
Category:
Company
Categories
Address:
9701 West Pico Blvd., #205
City:
Los Angeles
State:
CA
Zip Code:
90035
Phone:
310-552-2541
Fax:
310-552-2841
E-Mail:
cortex@cinenet.net

Fig. 14

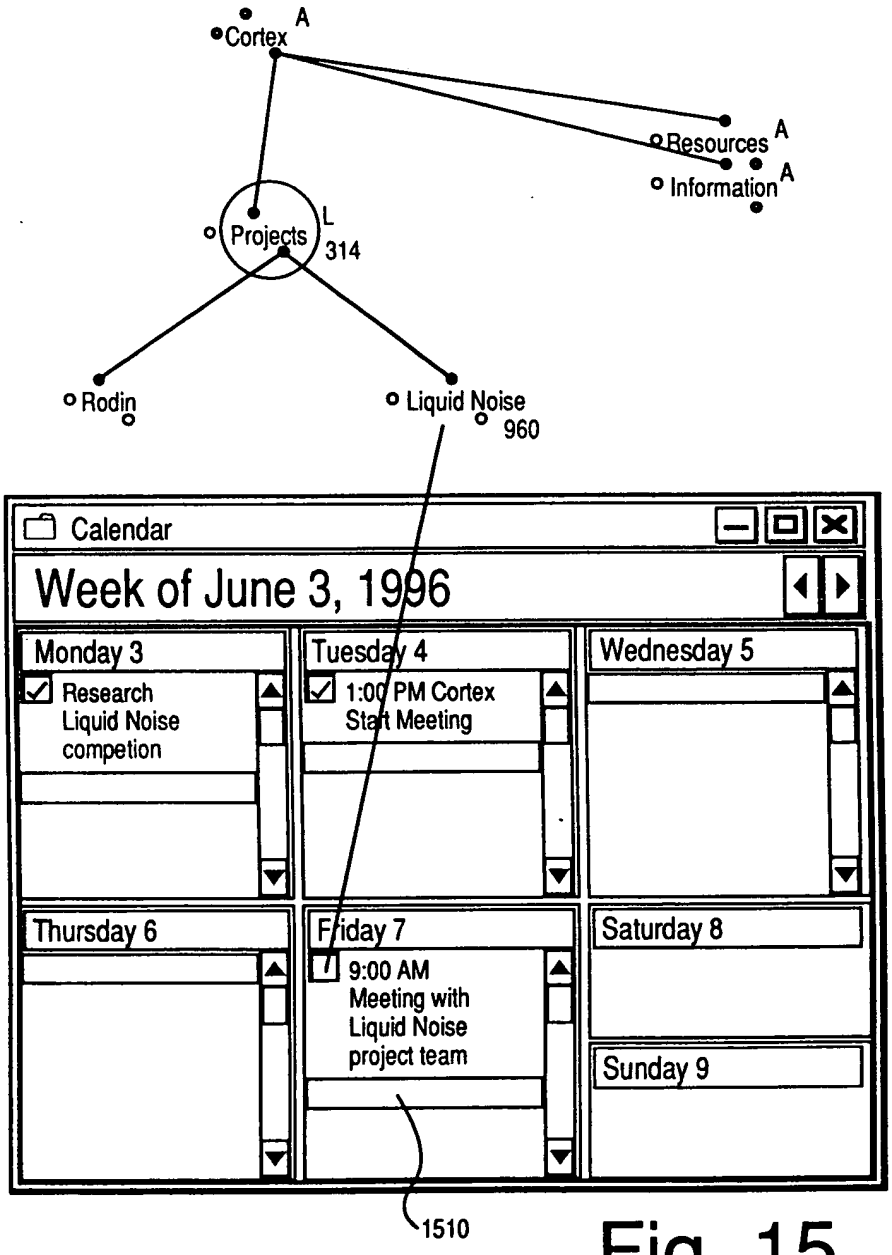
Cortex
A
Resources
A
Information
A
Projects
L
314
Rodin
Liquid Noise
960
Calendar
Week of June 3, 1996
Monday 3
Research
Liquid Noise
competion
Tuesday 4
1:00 PM Cortex
Start Meeting
Wednesday 5
Thursday 6
Friday 7
9:00 AM
Meeting with
Liquid Noise
project team
Saturday 8
Sunday 9
1510

Fig. 15

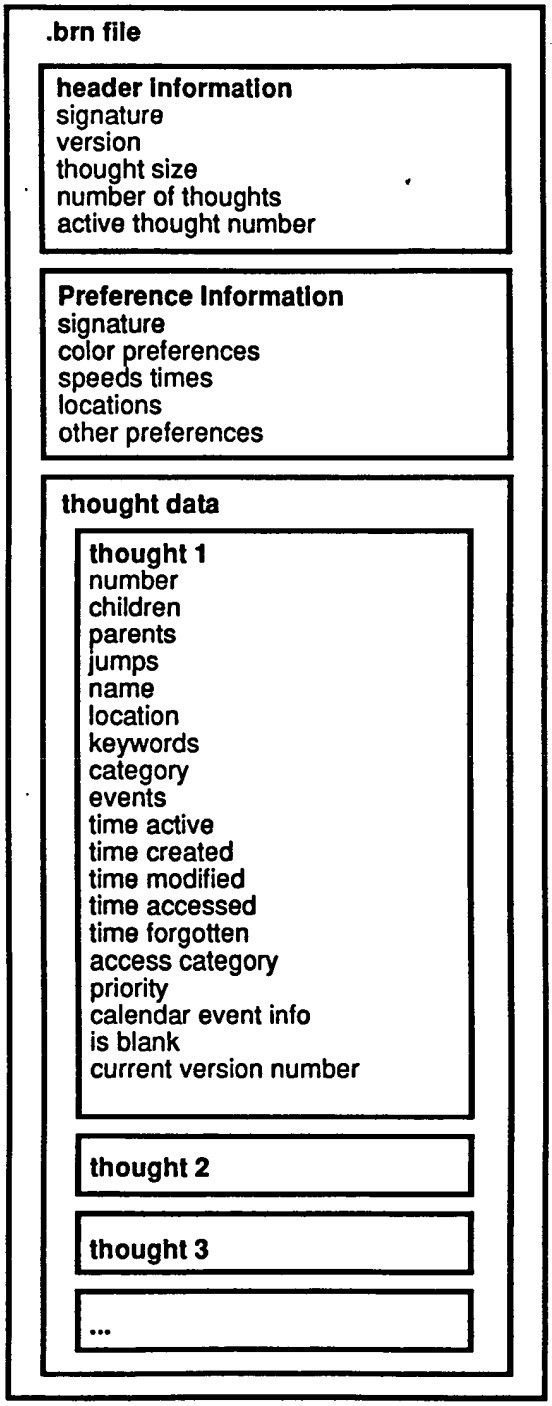
.brn file
header information
signature
version
thought size
number of thoughts
active thought number
Preference Information
signature
color preferences
speeds times
locations
other preferences
thought data
thought 1
number
children
parents
jumps
name
location
keywords
category
events
time active
time created
time modified
time accessed
time forgotten
access category
priority
calendar event info
is blank
current version number
thought 2
thought 3
...

Fig. 16

```
ForgetThought (fNum)
{
  // mark all the children of the selected thought
  list.Clear();
  MarkChildren(fNum, list);
  // unmark the active thought
  list.RemoveThought(activeThought);
  // unmark thoughts with unmarked parents
  lNum = list.GetFirstNum();
  while(lNum !=0)
  {
    if(lNum != fNum)  // don't unmark the selected thought
    {
      pNum = GetFirstThoughtParent(lNum);
      while(pNum != 0)
      {
        if(list.Contains(pNum) == FALSE)
        {
          if(IsThoughtInLongTermMemory(pNum)  == FALSE)
          {
            // unmark all the children of the unmarked parent
            childList.Clear();


            MarkChildren(pNum, childList);
            list.RemoveList(childList);
          }
        }
        pNum = GetNextThoughtParent(lNum);
      }
    }
    lNum = list.GetNextNum();
  }
  // now forget all the thoughts left on the list
  lNum = list.GetFirstNum();
  while(lNum != 0)
  {
    ForgetThought(lNum);
    lNum = list.GetNextNum();
  }
}

RememberThought(rNum)
{
  // mark all the children of the selected thought
  list.Clear();
  MarkChildren(rNum, list);
  // remember all the thoughts on the list
  lNum = list.GetFirstNum();
  while(lNum != 0)
  {
    RememberThought(lNum);
    lNum = list.GetNextNum();
  }
}

MarkChildren(num, list)
{
  list.AddThought(num);
  cNum = GetFirstChild(num);
  while(cNum != 0)
  {
    MarkChildren(cNum, list);
    cNum = GetNextChild(num);
  }
}
```

FIG. 17

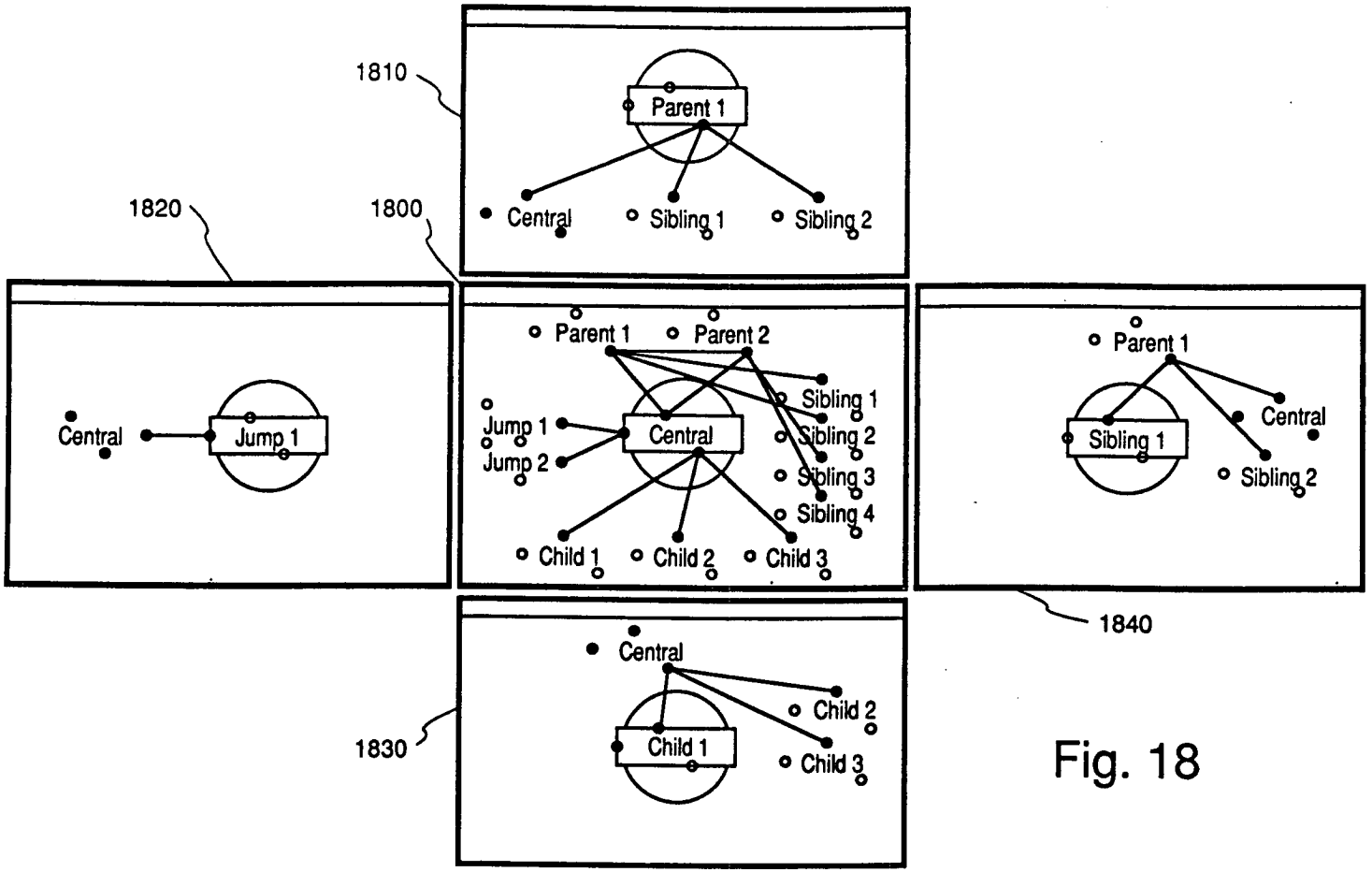
1810
Parent 1
Central
Sibling 1
Sibling 2
1820
1800
Central
Jump 1
Parent 1
Parent 2
Jump 1
Jump 2
Central
Sibling 1
Sibling 2
Sibling 3
Sibling 4
Child 1
Child 2
Child 3
Parent 1
Sibling 1
Central
Sibling 2
1840
Central
Child 1
Child 2
Child 3
1830

Fig. 18

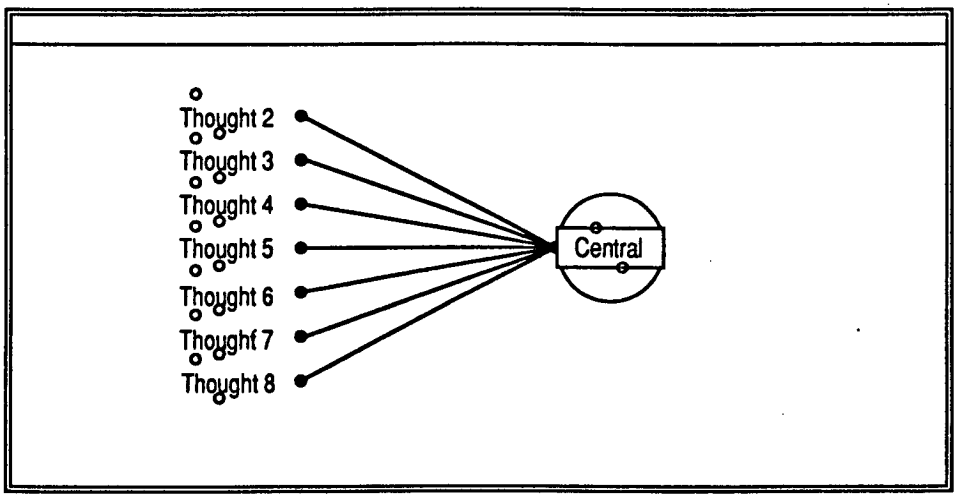
Thought 2
Thought 3
Thought 4
Thought 5
Thought 6
Thought 7
Thought 8
Central

Fig. 19

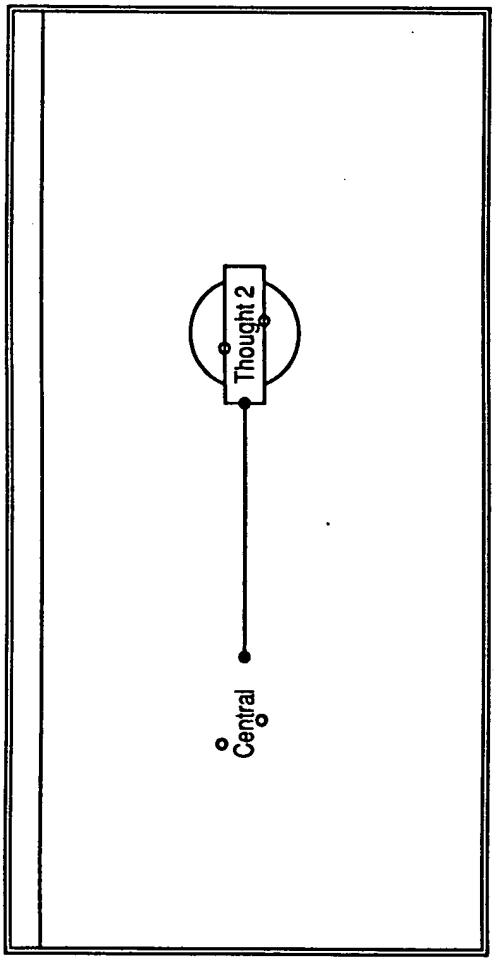

Fig. 20

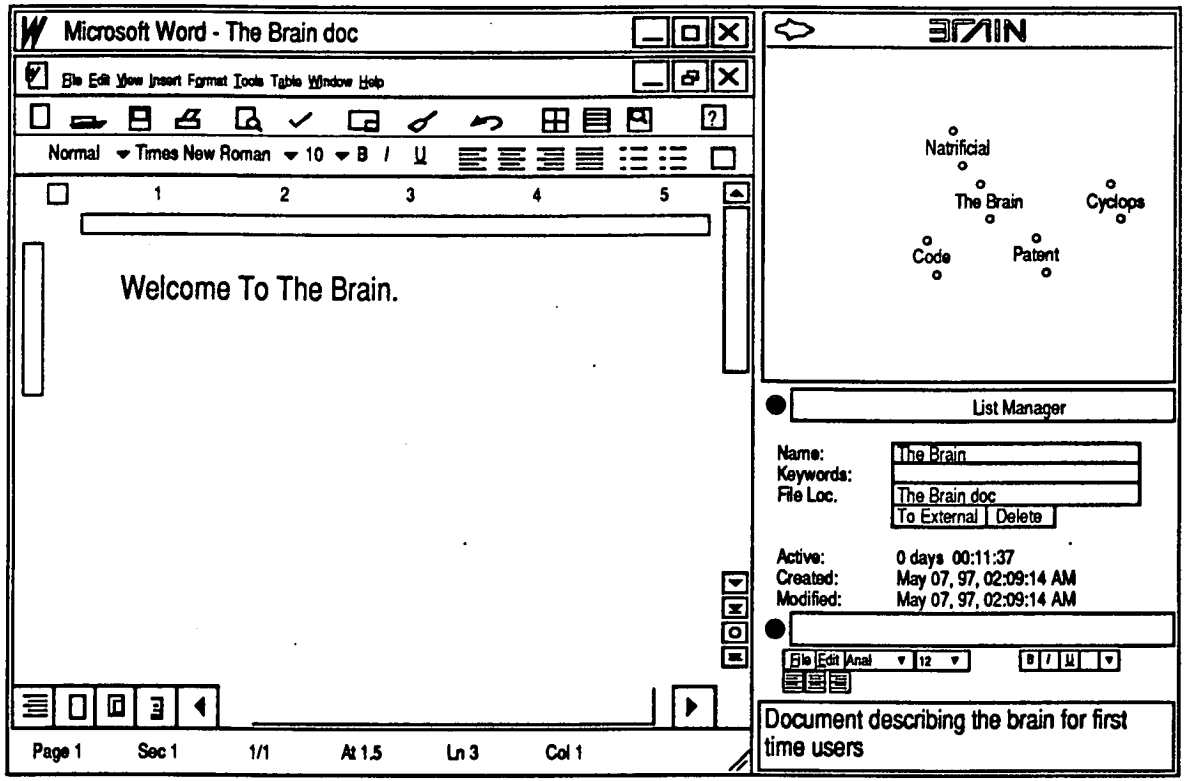
Microsoft Word - The Brain doc
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
10
Welcome To The Brain.
Page 1
Sec 1
1/1
At 1.5
Ln 3
Col 1
BRAIN
Natrificial
The Brain
Cyclops
Code
Patent
List Manager
Name:
Keywords:
File Loc.
The Brain
The Brain.doc
To External
Delete
Active:
Created:
Modified:
0 days 00:11:37
May 07, 97, 02:09:14 AM
May 07, 97, 02:09:14 AM
Document describing the brain for first time users

Fig. 21

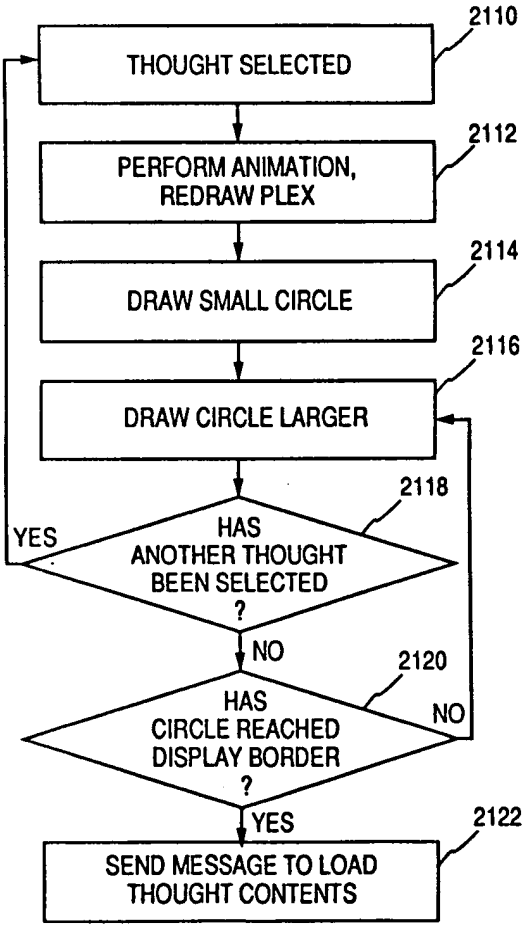
2110
THOUGHT SELECTED
2112
PERFORM ANIMATION, REDRAW PLEX
2114
DRAW SMALL CIRCLE
2116
DRAW CIRCLE LARGER
2118
HAS ANOTHER THOUGHT BEEN SELECTED ?
YES
NO
2120
HAS CIRCLE REACHED DISPLAY BORDER ?
NO
YES
2122
SEND MESSAGE TO LOAD THOUGHT CONTENTS

Fig. 22

Algorithm for drawing the plex with distant thoughts

```
1.      Create a list of thoughts to be drawn and their screen locations:
2.          Add the central thought to the list.
3.          Add children to the list.
4.          Add parents to the list.
5.          Add jumps to the list.
6.          Add siblings to the list, checking first that they
                are not already on the list.
7.          Add distants of children to the list, checking first that they
                are not already on the list.
8.          Add distants of parents to the list, checking first that they
                are not already on the list.
9.          Add distants of jumps to the list, checking first that they
                are not already on the list.
10.         Add distants of siblings to the list, checking first that
                they are not already on the list.
11.     Draw the lines that connect each thought:
12.         For each item in the list:
13.             Get each item in the list:
14.                 If the two items are related, draw lines
                        between them from and to the appropriate
                        gates.
15.     Draw the distant thoughts:
16.         For each item in the list:
17.             If it is a distant thought, draw it.
18.     Draw the other thoughts:
19.         For each item in the list:
20.             If it is not a distant thought, draw it.
```

Fig. 23

```
// the non recursive method for searching thoughts
// tries to find a route from nSrc to nDest other than a direct relation
// returns TRUE if found
boolean Search(int nSrc, int nDest)
{
     //create the lists
     ThoughtList posList;     //list of thoughts that possibly connect
     ThoughtList notList;     //list of thoughts that do not connect
     //empty the lists
     posList.Initialize();
     notList.Initialize();

     //add the source to the not list since we cannot go directly
     to the destination,
     notList.Add(nSrc);

     //since we cannot go directly to the destination,
     //add all relates except the destination to the possible list
     Thought src(nSrc);
     for(int n = 0;;n++)
     {
          int nRel = src.GetRelate(n);
          if(!nRel)
          {
               //no more relations, done
               break;
          }
          if(nRel != nDest)
          {
               //add it to the possibly connect list
               posList.Add(nRel);
          }
     }

     while(TRUE)
     {
          //check the first possibility
          int nTest = posList.GetFirst();
          if(!nTest)
          {
               //nothing on the list, done
               break;
          }
          Thought test(nTest);
          if(test.IsRelated(nDest))
          {
               //this one is related to the destination, we're done
               return TRUE;
          }
          // does not connect, add it to the does not connect list
          notList.Add(nTest);
          // add all related thoughts except those already checked to
          possible list for
          (int n = 0;; n++)
          {
               int nRel = test.GetRelate(n);
               if(!nRel)
               {
                              //no more relations, done
                              break;
               }
               if(!notList.Exists(nRel))
               {
                              //not checked yet, add to possible list
                              posList.Add(rel);
               }
          }
          //remove this one from the possible list
          posList.Remove(nTest);
     }
     //we've checked everything there is no other way to get from
     nSrc to nDest
     return FALSE;
}
```

Fig. 24

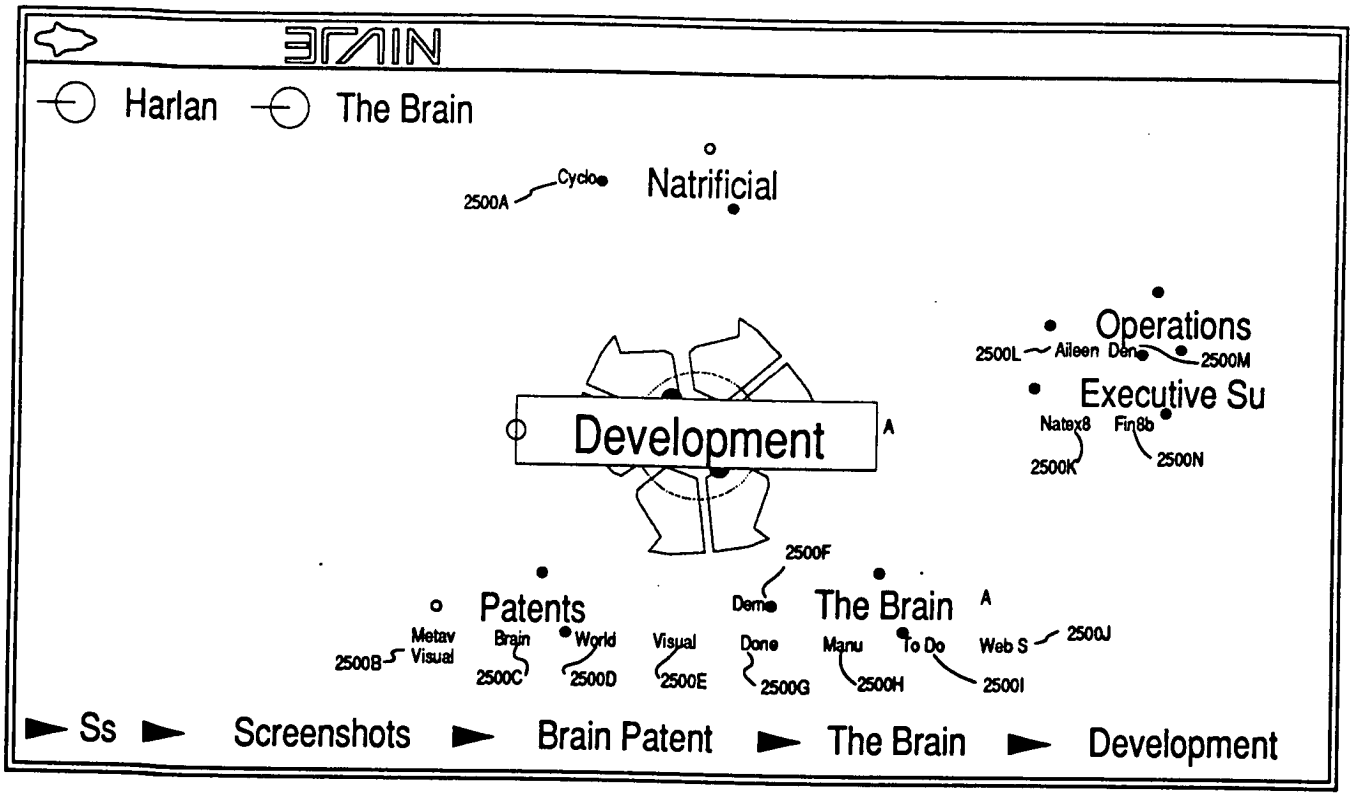
Harlan
The Brain
Natrificial
2500A
Cyclo
Development
A
Operations
2500L
Aileen
Den
2500M
Executive Su
Natex8
Fin8b
2500K
2500N
2500F
Patents
Demo
The Brain
A
Metav
Visual
2500B
Brain
2500C
World
2500D
Visual
2500E
Done
2500G
Manu
2500H
To Do
2500I
Web S
2500J
Ss
Screenshots
Brain Patent
The Brain
Development

Fig. 25

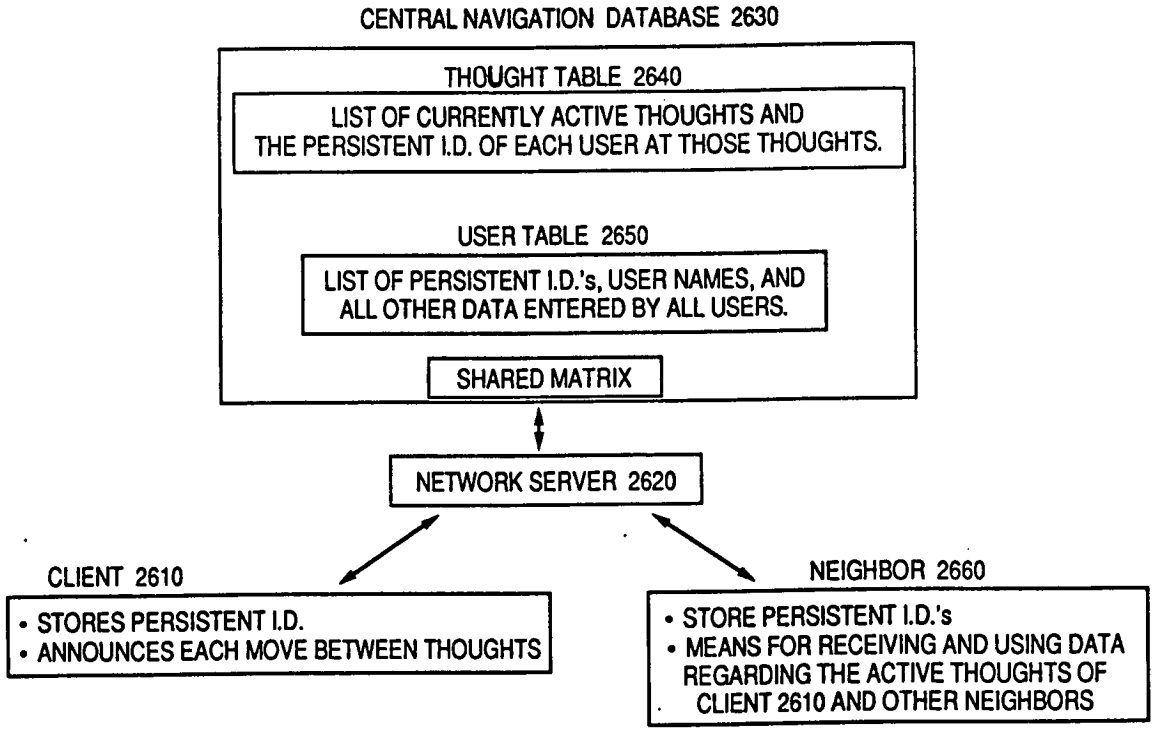
CENTRAL NAVIGATION DATABASE 2630
THOUGHT TABLE 2640
LIST OF CURRENTLY ACTIVE THOUGHTS AND THE PERSISTENT I.D. OF EACH USER AT THOSE THOUGHTS.
USER TABLE 2650
LIST OF PERSISTENT I.D.'s, USER NAMES, AND ALL OTHER DATA ENTERED BY ALL USERS.
SHARED MATRIX
NETWORK SERVER 2620
CLIENT 2610
• STORES PERSISTENT I.D.
• ANNOUNCES EACH MOVE BETWEEN THOUGHTS
NEIGHBOR 2660
• STORE PERSISTENT I.D.'s
• MEANS FOR RECEIVING AND USING DATA REGARDING THE ACTIVE THOUGHTS OF CLIENT 2610 AND OTHER NEIGHBORS

Fig. 26

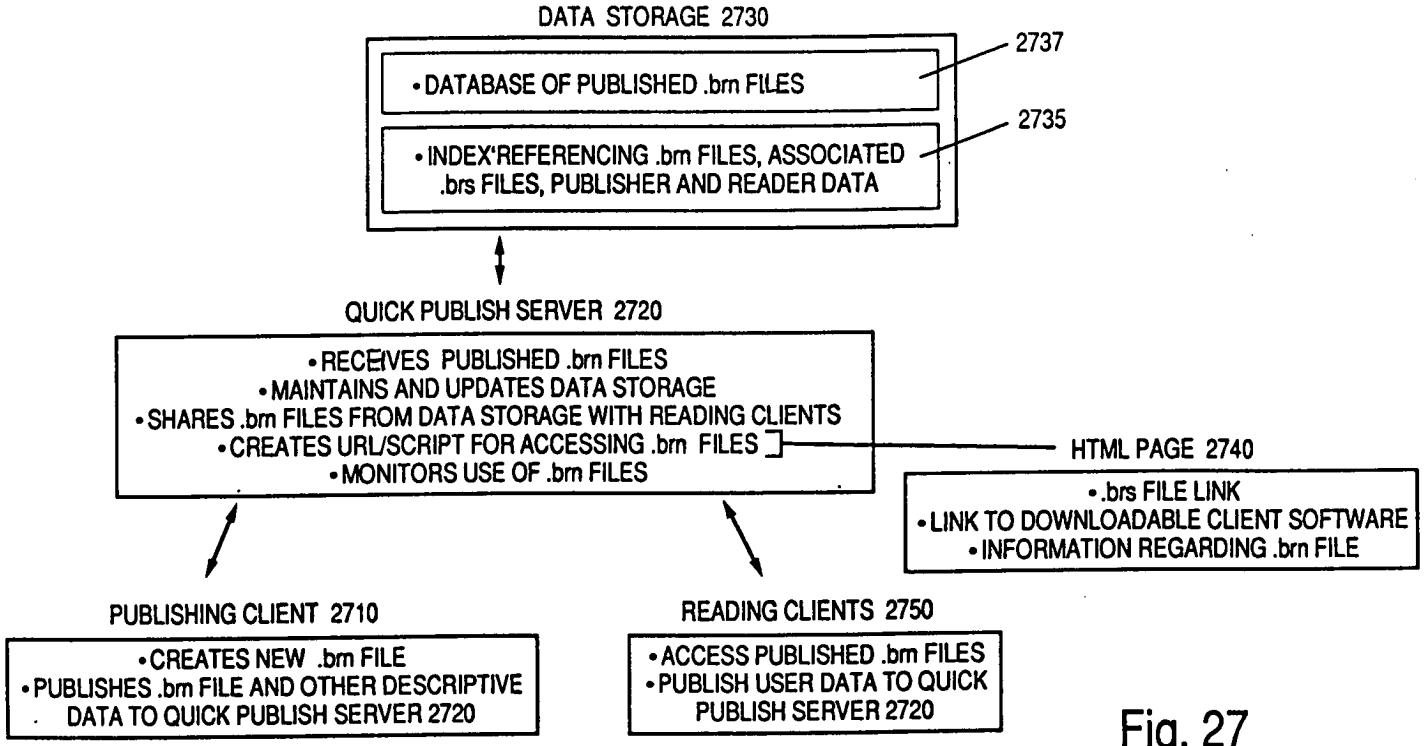
DATA STORAGE 2730
• DATABASE OF PUBLISHED .brn FILES
2737
• INDEX REFERENCING .brn FILES, ASSOCIATED .brs FILES, PUBLISHER AND READER DATA
2735
QUICK PUBLISH SERVER 2720
• RECEIVES PUBLISHED .brn FILES
• MAINTAINS AND UPDATES DATA STORAGE
• SHARES .brn FILES FROM DATA STORAGE WITH READING CLIENTS
• CREATES URL/SCRIPT FOR ACCESSING .brn FILES
• MONITORS USE OF .brn FILES
HTML PAGE 2740
• .brs FILE LINK
• LINK TO DOWNLOADABLE CLIENT SOFTWARE
• INFORMATION REGARDING .brn FILE
PUBLISHING CLIENT 2710
• CREATES NEW .brn FILE
• PUBLISHES .brn FILE AND OTHER DESCRIPTIVE DATA TO QUICK PUBLISH SERVER 2720
READING CLIENTS 2750
• ACCESS PUBLISHED .brn FILES
• PUBLISH USER DATA TO QUICK PUBLISH SERVER 2720

Fig. 27

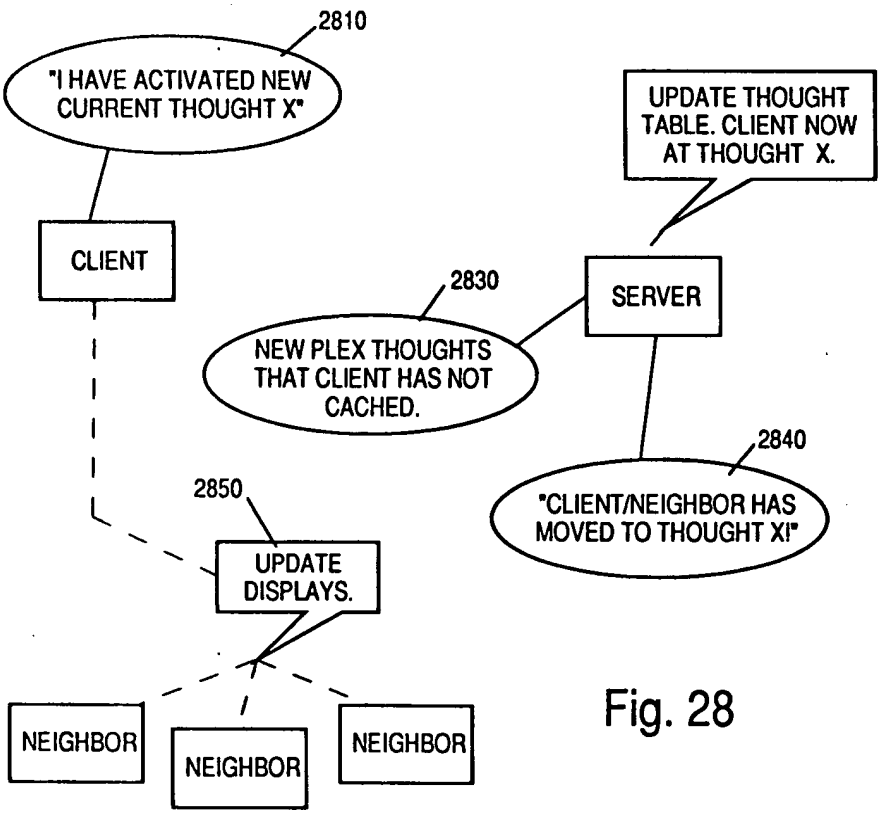
2810
"I HAVE ACTIVATED NEW CURRENT THOUGHT X"
UPDATE THOUGHT TABLE. CLIENT NOW AT THOUGHT X.
CLIENT
2830
SERVER
NEW PLEX THOUGHTS THAT CLIENT HAS NOT CACHED.
2840
"CLIENT/NEIGHBOR HAS MOVED TO THOUGHT X!"
2850
UPDATE DISPLAYS.
NEIGHBOR
NEIGHBOR
NEIGHBOR

Fig. 28

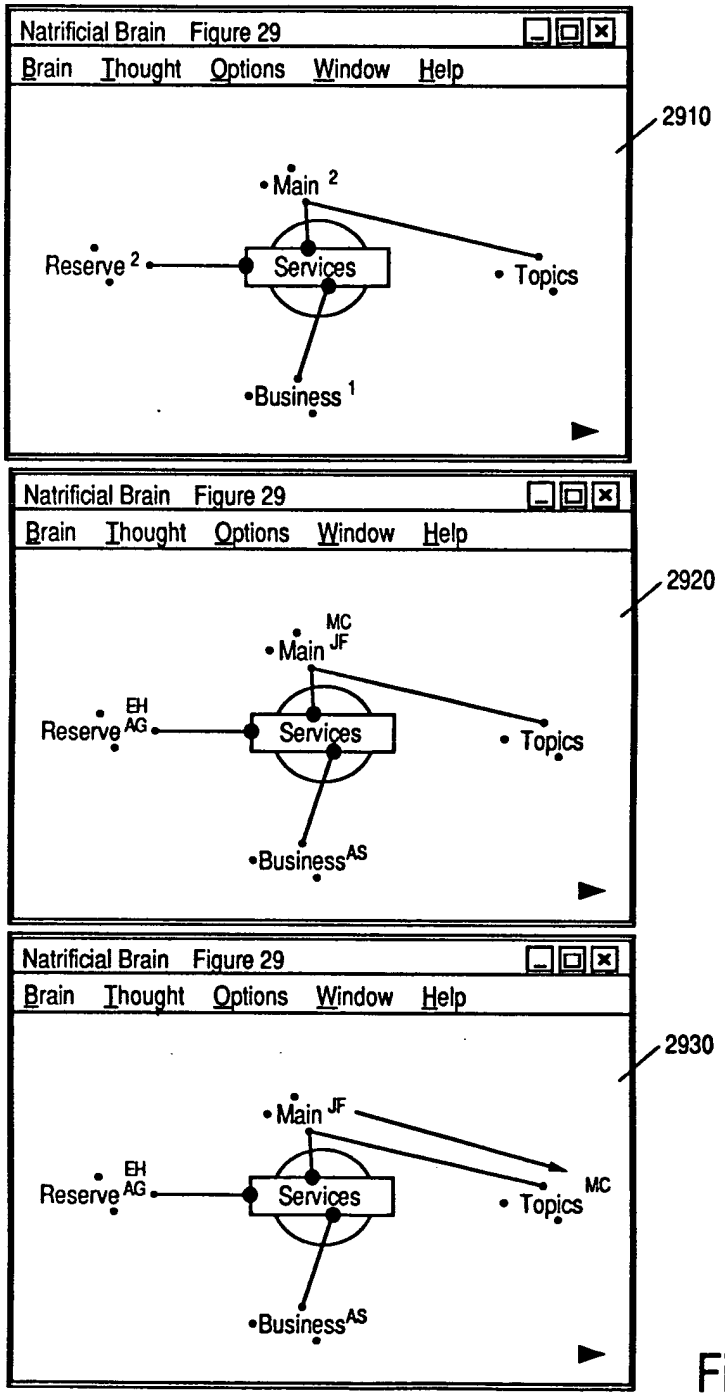
Natrificial Brain Figure 29
Brain Thought Options Window Help
2910
Main 2
Reserve 2
Services
Topics
Business 1
Natrificial Brain Figure 29
Brain Thought Options Window Help
2920
Main MC JF
Reserve EH AG
Services
Topics
Business AS
Natrificial Brain Figure 29
Brain Thought Options Window Help
2930
Main JF
Reserve EH AG
Services
Topics MC
Business AS

Fig. 29

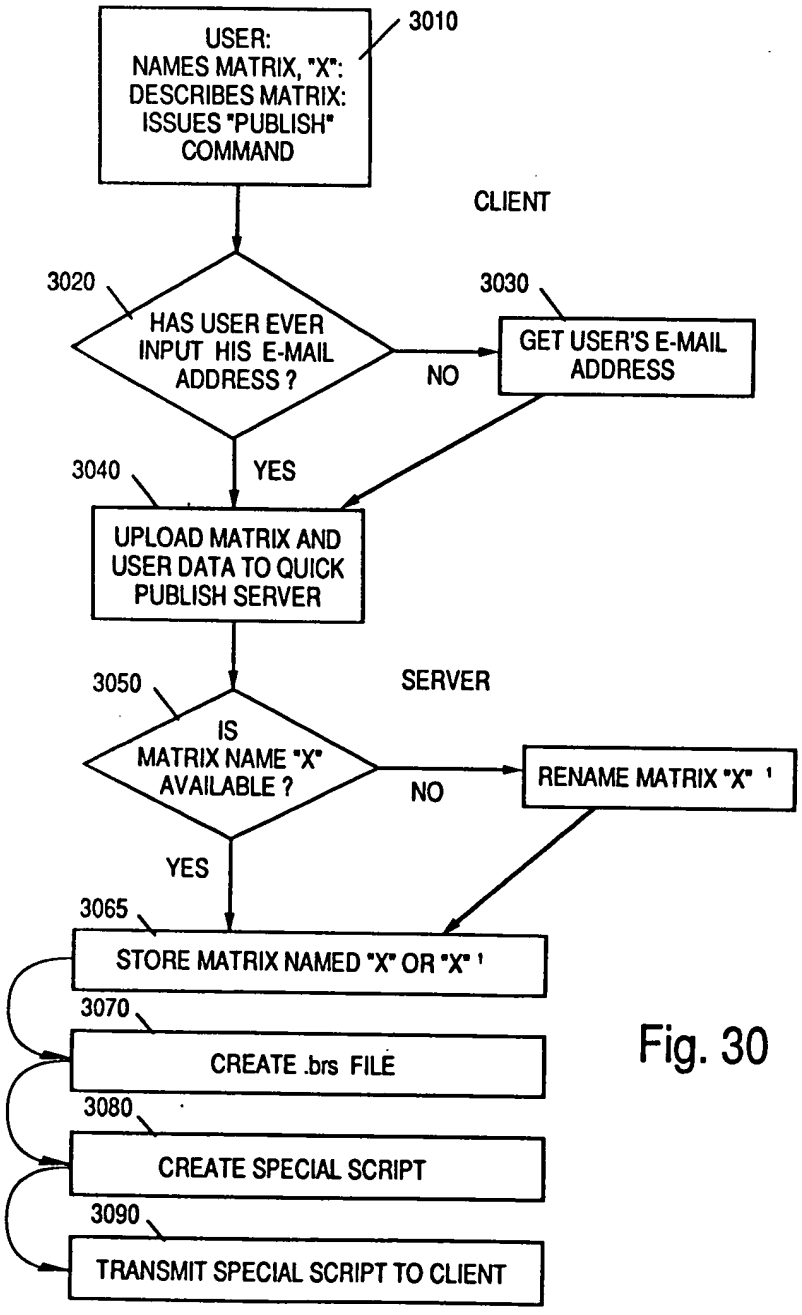
USER:
NAMES MATRIX, "X":
DESCRIBES MATRIX:
ISSUES "PUBLISH"
COMMAND
3010
CLIENT
3020
HAS USER EVER
INPUT HIS E-MAIL
ADDRESS ?
NO
3030
GET USER'S E-MAIL
ADDRESS
YES
3040
UPLOAD MATRIX AND
USER DATA TO QUICK
PUBLISH SERVER
SERVER
3050
IS
MATRIX NAME "X"
AVAILABLE ?
NO
RENAME MATRIX "X" '
YES
3065
STORE MATRIX NAMED "X" OR "X" '
3070
CREATE .brs FILE
3080
CREATE SPECIAL SCRIPT
3090
TRANSMIT SPECIAL SCRIPT TO CLIENT

Fig. 30

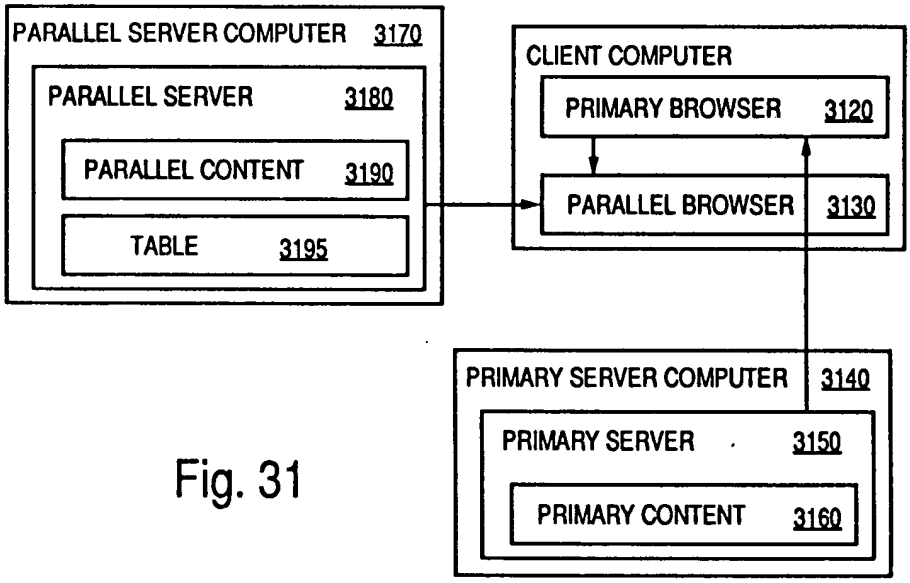
PARALLEL SERVER COMPUTER 3170
PARALLEL SERVER 3180
PARALLEL CONTENT 3190
TABLE 3195
CLIENT COMPUTER
PRIMARY BROWSER 3120
PARALLEL BROWSER 3130
PRIMARY SERVER COMPUTER 3140
PRIMARY SERVER 3150
PRIMARY CONTENT 3160

Fig. 31

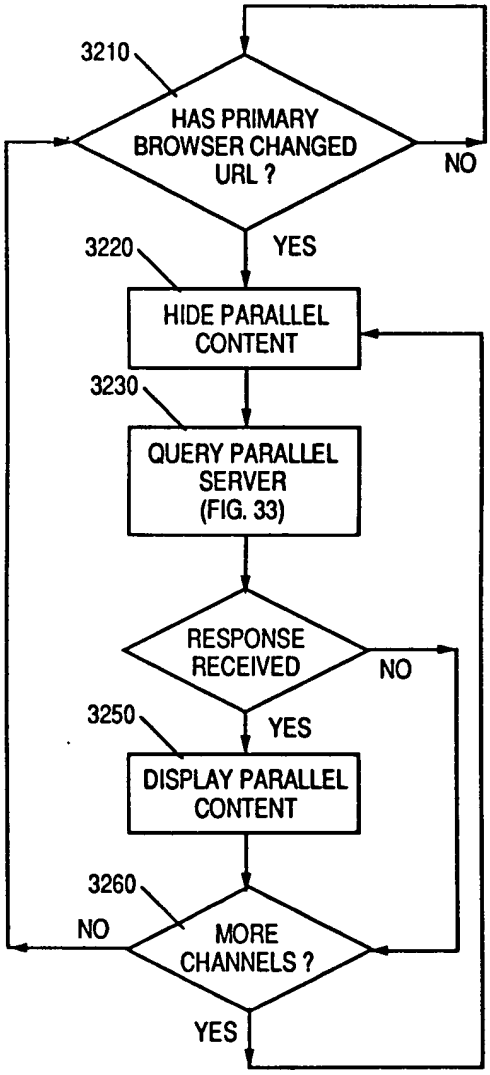
3210
HAS PRIMARY BROWSER CHANGED URL ?
NO
YES
3220
HIDE PARALLEL CONTENT
3230
QUERY PARALLEL SERVER (FIG. 33)
RESPONSE RECEIVED
NO
YES
3250
DISPLAY PARALLEL CONTENT
3260
NO
MORE CHANNELS ?
YES

Fig. 32

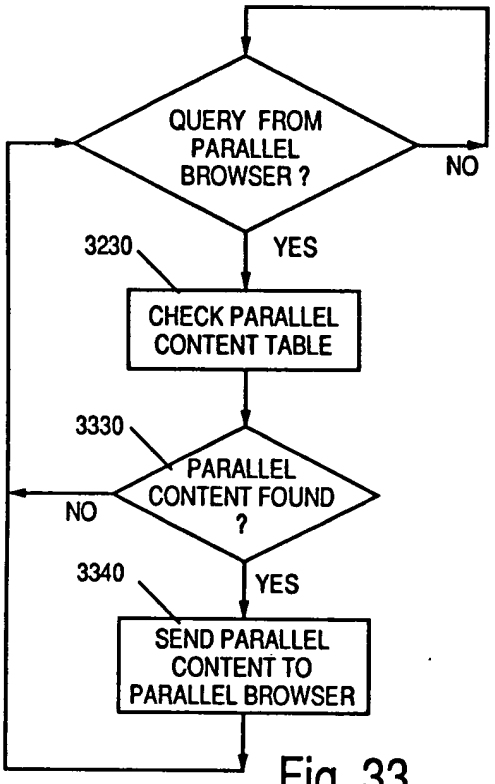
QUERY FROM PARALLEL BROWSER ?
NO
YES
3230
CHECK PARALLEL CONTENT TABLE
3330
PARALLEL CONTENT FOUND ?
NO
YES
3340
SEND PARALLEL CONTENT TO PARALLEL BROWSER

Fig. 33